17th International Conference on the Application of Computer
Science and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 12–14 July 2006

MINIMIZING THE TOTAL DISCOUNTED COST OF DISMANTLING A NUCLEAR POWER PLANT

J.–H. Bartels*, **and J. Zimmermann**

**Clausthal, University of Technology, Institute of Management and Economics*
Julius-Albert-Str. 2, D-38678 Clausthal-Zellerfeld, Germany
E-mail: jan-hendrik.bartels@tu-clausthal.de

Keywords: Phasing out of Nuclear Energy, Dismantling of Nuclear Power Plants, Resource-Constrained Project Scheduling, Net-Present-Value Optimization.

Abstract. *Due to economical, technical, or political reasons all over the world about 100 nuclear power plants have been disconnected until today. All these power stations are still waiting for their complete dismantling which, considering one reactor, causes costs of up to one Bil. Euros and lasts up to 15 years. In this paper we present a resource-constrained project scheduling approach minimizing the total discounted cost of dismantling a nuclear power plant. For the respective NP-hard optimization problem, we introduce an appropriate project scheduling model with minimum and maximum time lags, renewable and cumulative resources as well as multiple execution modes. Optimal solutions can be gained from a relaxation based enumeration approach which may be incorporated into a branch-and-bound algorithm.*

1 MOTIVATION

On April 26th, 2002 the German government passed a law with the purpose of phasing out of nuclear energy within three decades. In general, two facts are regulated by the law that is abbreviated by “AtG-E”. Firstly, it prescribes the remaining electrical power each of the 19 German nuclear power plants is allowed to produce until it is decommissioned (§1 (1) AtG-E), and secondly, it rules that contaminated material arising from the dismantling must be stored at the former power plant’s location until a national repository exists (§9a (2) AtG-E). In negotiations with the government, plant operators obtained the right to distribute the production of the remaining electrical power among their power stations, such that less efficient power plants can be disconnected earlier in order to enlarge the residual term for more efficient ones (§7 (1d) AtG-E). Nevertheless, in 2034 no nuclear power plant is allowed to operate anymore. However, due to the distribution of the remaining nuclear power, the first minor efficient power plants have already been disconnected, e.g. in 2003 the nuclear power plant in Stade was decommissioned. Further reactors, like in Greifswald or Rheinsberg, were previously taken out of operation because of economical and technical reasons. Actually, all over the world there exist about 100 power stations that have stopped producing nuclear energy and are still waiting for their complete dismantling. Considering that a dismantling project for one nuclear power plant causes cost of 500 Mio. to one Bil. Euros (cf. [6]), one can imagine that there exists a significant market for such projects and appropriate planning software.

In general, two common accepted variants of dismantling a reactor can be distinguished, which differ by the point of time the reactor is removed. In case of *immediate dismantling* the disassembling of the reactor is begun as soon as the power plant has been disconnected. In contrast, in case of *safe enclosure and postponed dismantling* one begins to remove the nuclear structure of the power station after it has been left as it was for a specified number of years (cf. [6]). The main argument for the latter case are lower costs today due to the application of discount rates. Moreover, lengthy delays decrease radioactivity over the time period such that the worker dose and public exposure are reduced. Nevertheless, postponed dismantling will require a continuing need for security, surveillance and maintenance. However, the direct dismantling is the active policy in many countries, e.g. Germany and Sweden, as it is in accord with the concept of a sustainable development. Regarding the ‘polluter pays’ principle, it prevents from inter-generational tensions (cf. [6]). Furthermore, this concept ensures that a high qualified staff for the challenging dismantling tasks is available, as nobody possesses better knowledge on the structures and facilities of a nuclear power plant than the staff that operated it or even was involved in its construction. Furthermore, we have to consider social aspects regarding the staff. As with phasing out of nuclear power in Germany there is little need for the highly specialized employees of the disconnected power plants, the only opportunity for keeping these employees occupied any longer is to involve them in an immediately starting dismantling project.

Although immediate dismantling compered to safe enclosure and postponed dismantling is disadvantageous in regard to discounted costs, there exist opportunities to reduce this disadvantage. In general, our approach makes use of the fact that the entire dismantling project can be subdivided into individual activities, each of which leading to a disbursement. Of course, the discounted cost of each activity depends on the date it is processed. Consequently, the total discounted costs of a dismantling project can be optimized by scheduling the activities in an appropriate manner, where a number of constraints must be regarded. The constraints are necessary due to technological and logical dependencies between several activities as well as scarce resources which are necessary to perform the individual activities. That is, we consider a resource-constrained project scheduling problem optimizing the net-present-value of the immediate dismantling. In the subsequent section, we present the optimization problem in detail and give an introduction to necessary concepts of resource-constrained project scheduling.

2 PROBLEM DESCRIPTION AND MODEL

Under long-term considerations, the dismantling of a nuclear power plant can be described as a project containing n *sub-projects* or *disassembling sections* each of which leading to a disbursement depending on the manner it is processed. For simplicity we call these sub-projects *activities*. These activities have to be scheduled, such that the resulting sum of discounted disbursements is minimized and a number of constraints are met. These constraints can be summarized as follows.

- *temporal constraints:* minimum and maximum time lags between activities must be regarded
- *mode constraints:* most activities can be executed in two modes –internal and external processing– that influence the disbursement of the activity
- *resource and inventory constraints:* each activity requires some scarce resources, where the requirement may depend on the mode the activity is executed in

Subsequently, we describe the scheduling problem under consideration in more detail and give an introduction to necessary terms and concepts of resource-constrained project scheduling. A detailed introduction to this terms and concepts is given by Zimmermann et al. [7]. Let $i = 1, \ldots, n$ be the activities of the project in question. By $p_i \in \mathbb{N}$ we denote the *duration* or *processing time* of activity i, which is assumed to be carried out without interruption. In addition, we introduce the *fictitious activities* 0 and $n + 1$ representing the beginning and completion, respectively, of the project, where $p_0 = p_{n+1} = 0$. Then $V = \{0, 1, \ldots, n + 1\}$ is the set of all activities.

Let $S_i \geq 0$ be the *start time* of activity $i \in V$, where we set $S_0 := 0$ (i.e. the project always begins at time zero). Then S_{n+1} represents the *project duration*. We assume that $S_{n+1} \leq \overline{d}$, where $\overline{d} \in \mathbb{N}$ is a prescribed maximum project duration. For the problem under consideration this maximum project duration is 12 to 15 years, depending on negotiations between the power station operator and legislator. A sequence $S = (S_0, S_1, \ldots, S_{n+1})$ with $S_i \geq 0$ $(i \in V)$ and $S_0 = 0$ is called a *schedule*.

Let us take a look at the temporal constraints between the project activities. A *minimum time lag* $d_{ij}^{min} \in \mathbb{Z}_{\geq 0}$ or *maximum time lag* $d_{ij}^{max} \in \mathbb{Z}_{\geq 0}$ can be prescribed between the start of two different activities i and j, that is, $S_j - S_i \geq d_{ij}^{min}$ or $S_j - S_i \leq d_{ij}^{max}$, respectively. By minimum time lags we model release dates for activities regarding the fact that some activities cannot start before the radioactivity has been decreased over the time period. Furthermore, it takes some time until dismantling licenses are granted which also leads to minimum time lags. For $d_{ij}^{min} = p_i$ we receive a special type of temporal constraint, called *precedence constraint*. Precedence constraints are frequently used for activities starting at the earliest after several preceding activities have been finished due to existing logical dependencies. Take a disassembling sequence for example. Maximum time lags are necessary in order to represent regulatory licenses' duration of validity to perform a disassembling section as well as to ensure that an activity starts as soon as a preceding activity has been finished, e.g. several dismantling tasks must immediately be followed by a decontamination activity. Finally, to ensure that the project is terminated by time $\overline{d}$, we introduce the maximum time lag $d^{max}_{0,n+1} := \overline{d}$.

It is well-known that an *activity-on-node network* N can be uniquely assigned to the project in question. To do so we identify the activities $i \in V$ with the nodes of network N. If there is a minimum time lag d^{min}_{ij}, we introduce an arc $\langle i, j\rangle$ with weight $\delta_{ij} := d^{min}_{ij}$. If there is a maximum time lag d^{max}_{ij}, we introduce an arc $\langle j, i\rangle$ with weight $\delta_{ji} := -d^{max}_{ij}$. Due to maximum time lags, network N generally contains cycles. Let E be the set of arcs of network N with $m =\mid E \mid$. Then the above inequalities for the minimum and maximum time lags can be summarized as

$$S_j - S_i \geq \delta_{ij} \quad (\langle i, j\rangle \in E) \tag{1}$$

representing all kinds of temporal constraints. A schedule S that satisfies this constraints (1) is termed *time-feasible*. The set of time-feasible schedules is denoted by $\mathcal{S}_T$. It holds that $\mathcal{S}_T \neq \emptyset$ exactly if network N does not contain any cycle of positive length (cf. [1]).

For many activities i there exists a set $M := \{0, 1\}$ of two alternative execution modes m_i (execution by *internal* ($m_i = 0$) or *external* ($m_i = 1$) staff). However, not all activities can be performed by an external service provider such that $M_i \subseteq M$ denotes the set of admissible modes for an activity i and a sequence $\mathcal{M} = (m_0, m_1, \ldots, m_{n+1})$ with $m_i \in M_i$ $(i \in V)$ and $m_0 = m_{n+1} = 0$ is called a *mode assignment*. An introduction to multi-mode project scheduling can be found in Heilmann [3]. The selected mode m_i influences the disbursement c_{im_i} of an activity i but not its processing time p_i, as the latter is fix due to technological reasons. Generally, engaging an external staff leads to a higher disbursement than using internal employees, i.e. $c_{i1} \geq c_{i0}$.

Carrying out the activities of the underlying project requires scarce resources $k \in \mathcal{R}^\rho$ like manpower, machines, special equipment or perhaps an amount of space in the dry well. As the available capacity $R_k \in \mathbb{N}$ of all these resources in each time period is independent from its use in former periods, they are called *renewable resources* (cf. [4]). Moreover, we assume the capacity of external service providers not being scarce. Then, let $r_{im_ik} \in \{0, 1, \ldots, R_k\}$ be the amount of resource k used by activity i if it is performed in mode m_i. Keep notice that, in general, only the requirement for manpower varies with the underlying mode, as the utilization of machines and equipment as well as the necessary amount of space in the dry well are independent from internal or external processing. Given schedule $S = (S_i)_{i\in V}$,

$$\mathcal{A}(S, t) := \{i \in V \mid S_i \leq t < S_i + p_i\}$$

is the set of activities in progress, also called the *active set*, at time $t \in [0, \overline{d}]$. Furthermore, given a mode assignment $\mathcal{M} = (m_i)_{i\in V}$ in addition to schedule S,

$$r_k(\mathcal{M}, S, t) := \sum_{i\in\mathcal{A}(S,t)} r_{i0k} \cdot (1 - m_i) + r_{i1k} \cdot m_i$$

is the amount of resource $k \in \mathcal{R}^\rho$ used at time $t \in [0, \overline{d}]$, such that the *renewable-resource constraints* are

$$r_k(\mathcal{M}, S, t) \leq R_k \quad (k \in \mathcal{R}^\rho,\ 0 \leq t \leq \overline{d}). \tag{2}$$

In dismantling projects of nuclear power plants, we make use of an additional type of scarce resource different from renewable ones. As mentioned in Section 1, contaminated material arising from the dismantling must be stored at the former power station's location. For this purpose an interim storage facility with sufficient capacity must be built. An additional storage is necessary to keep material from disassembling activities until it can be processed or recycled by a processing activity. Usually, the turbine house can be used for this purpose after its interior has been cleared out. Resources that are depleted and replenished during the project are called *cumulative resources* (cf. [4]). The availability of a cumulative resource k at a given time t results from all positive and negative requirements (depletions and replenishments, respectively) that have occurred by time t. The inventory level in resource k is supposed to be bounded from below by $\underline{R}_k = 0$ and from above by the capacity of the storage facility $\overline{R}_k \in \mathbb{Z}_{\geq 0}$.

Let $\mathcal{R}^\gamma$ be the set of all cumulative resources and let $r_{ik} \in \mathbb{Z}$ denote the inventory change of resource k caused by activity i (notice that r_{ik} does not depend on the mode activity i is executed in). If $r_{ik} > 0$, activity i replenishes resource k by r_{ik} units, and if $r_{ik} < 0$, resource k is depleted by $-r_{ik}$ units. We assume that resources $k \in \mathcal{R}^\gamma$ are depleted at completion times and replenished at start times of activities. To simplify writing, we also suppose that an activity cannot deplete and replenish one and the same cumulative resource. The initial stock r_{0k} of all cumulative resources k is set to $\overline{R}_k$ illustrating that the respective storage facility is not available at the project's start. The activities i representing the construction of the interim storage facility and the close up of the turbine house's interior, deplete the respective storage k completely (i.e. $r_{ik} = -\overline{R}_k$) such that the storage becomes available. Accordingly, at activity i representing the demolition of the turbine house, the respective storage k is replenished by $r_{ik} = \overline{R}_k$ units and no further material can be stored afterwards. Furthermore, all disassembling tasks replenish a storage, whereas activities which process the disassembled material deplete it.

Now let $V_k^- := \{i \in V \mid r_{ik} < 0\}$ and $V_k^+ := \{i \in V \mid r_{ik} > 0\}$ denote the sets of all activities $i \in V$ depleting and replenishing, respectively, resource k. For given schedule S

$$\mathcal{A}_k(S,t) := \{i \in V_k^- \mid S_i + p_i \leq t\} \cup \{i \in V_k^+ \mid S_i \leq t\}$$

is the *active set* of all activities that determine the inventory level

$$r_k(S,t) := \sum_{i \in \mathcal{A}_k(S,t)} r_{ik}$$

in resource k at time t. The *inventory constraints* can now be written as

$$\underline{R}_k \leq r_k(S,t) \leq \overline{R}_k \quad (k \in \mathcal{R}^\gamma,\ 0 \leq t \leq \overline{d}). \tag{3}$$

A schedule S that satisfies (2) is called (*renewable*-)*resource-feasible* and a schedule S which satisfies inventory constraints (3) is termed *inventory-feasible*.

A time-, resource- and inventory-feasible schedule is termed *feasible*. The set of feasible schedules is denoted by $\mathcal{S}$.

The underlying project scheduling problem consists of minimizing objective function $f(S, \mathcal{M})$ on the set $\mathcal{S}$ of feasible schedules. In detail, this problem reads as follows:

$$\left.\begin{array}{lll} \text{Min!} & f(S,\mathcal{M}) := \sum\limits_{i\in V} (c_{i0}\cdot(1-m_i) + c_{i1}\cdot m_i)\cdot(1+r)^{-S_i} & \\ \text{s. t.} & S_j - S_i \geq \delta_{ij} & (\langle i,j\rangle \in E) \\ & S_0 = 0 & \\ & m_i \in M_i & (i \in V) \\ & r_k(\mathcal{M}, S, t) \leq R_k & (k \in \mathcal{R}^\rho,\ 0 \leq t \leq \overline{d}) \\ & \underline{R}_k \leq r_k(S,t) \leq \overline{R}_k & (k \in \mathcal{R}^\gamma,\ 0 \leq t \leq \overline{d}) \end{array}\right\} \quad \text{(P)}$$

Given an interest rate r, the objective function $f(S, \mathcal{M})$ of problem (P) claims to minimize the total discounted disbursements with respect to the temporal, mode, resource and inventory constraints. As (P) is a generalization of the resource-constrained project scheduling problem with maximum time lags and antiregular objective function (cf. [4]), it can easily be shown that finding a feasible solution for (P) is NP-hard in the strong sense.

3 RELAXATION BASED SCHEDULE-GENERATION SCHEME

For solving project scheduling problem (P) a so-called *relaxation-based approach* turns out to be expedient, which has been devised by De Reyck and Herroelen [2]. If we omit the resource and inventory constraints (2) and (3) from problem (P), the resulting *resource relaxation* (RP) has feasible region $\mathcal{S}_T$ instead of $\mathcal{S}$, i.e. only the temporal constraints (1) must be considered. If we additionally tighten the mode constraints fixing the mode of internal processing for each activity, the objective function of (P) becomes antiregular, i.e. the objective function value decreases with increasing start times S_i of activities $i \in V$. For antiregular functions f, the *latest schedule* LS (i.e. the vector of latest start times LS_i of activities $i \in V$), which represents a maximal point of polytope $\mathcal{S}_T$, is optimal for (RP). This is due to the fact that internal processing leads to a smaller disbursement than engaging an external service provider ($c_{i0} \leq c_{i1}$) and the discounted disbursement of an activity decreases with increasing S_i. The latest schedule LS can efficiently be calculated by a label correcting algorithm (cf. [4]), which requires $\mathcal{O}(mn)$ time.

If schedule LS is resource- and inventory-feasible, we have found an optimal solution. Otherwise, there exist points in time t where the resource or inventory constraints (2) and (3) are violated. We distinguish three types of conflicts with renewable and cumulative resources.

Mode-independent resource conflict: for renewable resources k like machines, equipment or space in the dry well the used amount at time t does not depend on the selected mode m_i for any activity $i \in V$ and exceeds the maximum resource capacity R_k.

Mode-dependent resource conflict: like the previous resource conflict, but the resource usage depends on the selected modes for some activities $i \in V$ (e.g. manpower).

Inventory conflict: we speak of an *inventory excess* if the inventory in some resource k exceeds the maximum inventory level. One can imagine that generally also a situation occurs, where the inventory in some resource k falls below a minimum inventory level. In the problem under consideration this situation can be excluded because of an existing precedence constraint $\langle i, j\rangle$ between each inventory replenishing (disassembling) task i and the corresponding inventory depleting activity j with $r_{jk} = -r_{ik}$.

If a mode-independent (mode-dependent) resource conflict occurs, not all activities from active set $\mathcal{A}(S,t)$ can be processed at the same time (or in the selected mode). A set F of activities that cannot be executed simultaneously (in the selected mode), because

$$\sum_{i \in F} r_{i0k} \cdot (1 - m_i) + r_{i1k} \cdot m_i > R_k \text{ for some } k \in \mathcal{R}^\rho,$$

is called a *forbidden set* (*mode forbidden set*).

It has been shown by Bartusch et al. [1] that a schedule S is resource-feasible precisely if for any inclusion-minimal forbidden set F there exist two activities i, j such that activity j is completed before activity i starts, i.e. $S_j + p_j \leq S_i$ (we then say that forbidden set F has been broken up). This means that in case of a mode independent resource conflict at time t, we have to partition forbidden active set $\mathcal{A}(S,t)$ into two sets A and B such that B contains an activity j from each inclusion-minimal forbidden set $F \subseteq \mathcal{A}(S,t)$. If we then introduce the precedence constraints

$$S_i - S_j \geq p_j \quad (j \in B) \tag{4}$$

between some activity $i \in A$ and all activities $j \in B$, we break up all inclusion-minimal forbidden sets $F \subseteq \mathcal{A}(S,t)$ containing activity $i \in A$.

Since this property holds for any partition $\{A, B\}$ of $\mathcal{A}(S,t)$, without loss of generality we may restrict ourselves to inclusion-minimal sets B, which are referred to as *minimal delaying alternatives* in literature. An efficient recursive procedure for computing all minimal delaying alternatives for a given forbidden set F can be found in Neumann et al. [4]. We note that set A is a non-forbidden set and thus all activities $i \in A$ may be processed jointly.

Treating the mode-dependent resource conflicts is a little bit harder, as in addition to the precedence constraints (4) also changes in the mode assignment become appropriate to break up a mode forbidden set. Since external processing –if possible– requires no internal manpower and the resource capacity of external service providers is assumed not to be scarce, a mode forbidden set is broken up, if each activity $j \in B$ is either processed external or completed before some activity $i \in F \setminus B$. That is, for each minimal delaying alternative B and some activity $i \in F \setminus B$ we receive up to $2^{|B|}$ possibilities to break up the underlying mode forbidden set.

In the case of an inventory excess for a cumulative resource k at time t

$$\sum_{i \in \mathcal{A}_k(S,t)} r_{ik} > \overline{R}_k$$

holds true and $\mathcal{A}_k(S,t)$ is termed a *k-surplus set F*. Similarly to the approach for mode-independent resource conflicts, we introduce a concept of minimal *inventory forbidden sets*, which will allow us to define appropriate minimal delaying alternatives for resolving an inventory excess. In contrast to the case of renewable resources, our minimality concept cannot refer to inclusion-minimality, because resource demands may be positive or negative. That is why we define a k-surplus set F to be a *minimal k-surplus set* if no replenishing activity can be removed from F and no depleting activity can be added to F without loosing the surplus property. In Neumann et al. [4] it is shown that a schedule S is inventory-feasible if and only if each minimal k-surplus set F contains a depleting activity $i \in V_k^-$ and a replenishing activity $j \in V_k^+$ such that activity i is completed before activity j has been started (i.e. $S_i + p_i \leq S_j$). The corresponding minimal forbidden set satisfying this condition is said to be broken up in schedule S.

Now consider an inventory excess for some resource $k \in \mathcal{R}^\gamma$ at time t. A minimal delaying alternative B for active k-surplus set $\mathcal{A}_k(S,t)$ is an inclusion-minimal subset of $\mathcal{A}_k(S,t)$ containing a replenishing activity j of each minimal k-surplus set F that can be obtained from $\mathcal{A}_k(S,t)$ by deleting replenishing and adding depleting activities. By introducing the precedence constraints

$$S_j - S_i \geq p_i \quad (j \in B)$$

between some activity $i \in A = V_k^- \setminus \mathcal{A}_k(S,t)$ (i.e. a depleting activity i being completed after time t) and all activities $j \in B$, we break up all those of the above minimal k-surplus sets F containing i.

The relaxation-based approach for problem (P) is now as follows. We start by computing the latest schedule LS for the resource relaxation (RP) and choose internal processing for each activity $i \in V$ (i.e. $\mathcal{M} = (0)_{i \in V}$). If LS is resource- and inventory-feasible, we have found an optimal schedule. Otherwise, we determine some activity start time t for that schedule LS causes a resource or inventory conflict. By highest priority we break up all mode-independent resource conflicts. If actually no such conflicts occurs anymore, an excess of the inventory is resolved, and finally, if actually no mode-independent resource conflict or inventory excess exists, mode-dependent resource conflicts are treated. Within the same type of conflict that one, which occurs at latest time t, is resolved first. For breaking up forbidden sets $\mathcal{A}(S,t)$ or $\mathcal{A}_k(S,t)$, respectively, we compute a minimal delaying alternative B for F and refine relaxation (RP) by the corresponding precedence constraints (4) or, in the case of a mode-dependent resource conflict, also by altering the mode assignment. Each possibility to resolve an inventory or resource conflict leads to a (descending) node EN in the generated enumeration tree. We choose one of theses descending nodes EN and re-perform the determination of the latest schedule LS, which either shows the refined relaxation to be unsolvable, because we have generated a cycle of positive length in network N that was augmented by arcs $\langle i,j \rangle$ representing the additional precedence constraints, or which yields a new schedule S. We re-iterate these steps until we have reached a deadlock (i.e. N contains a cycle of positive length) or a feasible schedule S has been found. In doing so we perform a depth first search. As soon as we obtain a deadlock or a feasible schedule, we go back to the last node in our enumeration tree for that not all descendants have been examined yet and continue the enumeration, i.e. we perform backtracking.

4 TRUNCATED BRANCH AND BOUND ALGORITHMS

The relaxation based schedule-generation scheme introduced in Section 3 is expanded to a branch-and-bound algorithm. In each iteration, we branch over all pairs (i,B) for which B is a minimal delaying alternative for set $\mathcal{A}(S,t)$ or $\mathcal{A}_k(S,t)$ and i is some activity from set $A = \mathcal{A}(S,t) \setminus B$ or $A = \mathcal{A}_k(S,t) \setminus B$, respectively. Remember, that in the case of a mode-dependent resource conflict, for each pair (i,B) we additionally have to branch over all combinations resulting from altered mode assignments and added precedence relations. As usual, if in a node of the enumeration tree an appropriate lower bound LB on objective function value $f(S,\mathcal{M})$ (e.g. $f(LS,\mathcal{M})$ of the resource relaxation (RP) considered in the actual node) exceeds an upper bound UB on $f(S,\mathcal{M})$, we can stop branching the examined node. Furthermore, we reduce the number of nodes being examined by adding additional temporal constraints within a preprocessing procedure and by applying dominance rules in order to fathom enumeration nodes that are dominated by some nodes which have already been examined (cf. [5]). The introduced approach is promising to generate exact solutions for problem instances containing up to 30 activities. For larger problem instances we propose truncated versions of the branch-and-bound algorithm.

We receive a solution with performance guarantee ϵ if we stop branching a node EN in the enumeration tree as soon as for the lower and upper bound $(1+\epsilon)\cdot LB(EN) \geq UB$ (instead of $LB(EN) \geq UB$) holds true. A heuristic A is said to have a *performance guarantee* ϵ if $\frac{f_A(PI)-f^*(PI)}{f^*(PI)} \leq \epsilon$ for all problem instances PI and provided that $f^*(PI) > 0$, where $f_A(PI)$ is the objective function value for a feasible solution of PI computed by algorithm A and $f^*(PI)$ is the minimum objective function value for instance PI. Such a heuristic A is called a *performance-guaranteed algorithm*.

Another truncation of the branch-and-bound algorithm is received by sketching it to a *filtered beam search procedure* (cf. e.g. [4]). By φ and $\beta < \varphi$ we denote the integers corresponding to the *filter width* and the *beam width*, respectively. When we branch from an enumeration node P, only for the φ descending nodes EN' that have smallest objective function value f, lower bound $LB(EN')$ is calculated, and only the β nodes with smallest lower bounds are added to the enumeration tree. In contrast to the performance guaranteed algorithm, we may influence the depth of the enumeration tree for a filtered beam search algorithm, but neither a performance guarantee nor the computation of a feasible schedule in the case of $\mathcal{S} \neq \emptyset$ is ensured.

REFERENCES

[1] M. Bartusch , R. H. Möhring and F. J. Radermacher, Scheduling Project Networks with Resource Constraints and Time Windows. *Annals of Operations Research* **16**, 201–240, 1988.

[2] B. De Reyck and W. S. Herroelen, A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. *European Journal of Operational Research* **111**, 152–174, 1998.

[3] R. Heilmann, A Branch-and-Bound Procedure for the Multi-Mode Resource-Constrained Project Scheduling Problem with Minimum and Maximum Time Lags. *European Journal of Operational Research* **144**, 348–365, 2003.

[4] K. Neumann, C. Schwindt and J. Zimmermann, *Project Scheduling with Time Windows and Scarce Resources*. Springer, Berlin, 2003.

[5] K. Neumann and J. Zimmermann, Exact and Truncated Branch-and-Bound Procedures for Resource-Constrained Project Scheduling with Discounted Cash Flows and General Temporal Constraints. *Central European Journal of Operations Research* **10**, 357–380, 2002.

[6] L. Thomson, How is the international nuclear community addressing the issue of nuclear decommissioning? Is there a need for regulation on an international basis? *CEPMLP Annual Review* **3**, Article 5, 1999.

[7] J. Zimmermann, C. Stark and J. Rieck, *Projektplanung – Modelle, Methoden, Management*. Springer, Berlin, 2005.